Writing to a File

This tutorial describes three different ways of writing to a file: a write to a CF netCDF file, a single write to an IOAPI file, and multiple writes to an IOAPI file.

1. Writing to a CF netCDF file

The following opens a CF netCDF file for writing and writes to it:

```
## Open a file for writing,
## default format is CF netCDF
h = ioT.open("~/tmp/test.nc","w")

## Write a single variable to the file
h.write(o3)

## Write a list of variables to the file
h.write([co, no2])

## Close the file
h.close()
```

This assumes that you have already extracted the variables from other files (see "Variables" tutorial).

. Contents

2. Writing to an IOAPI file: Single write

The steps needed to open and write to an IOAPI file:

```
## open an IOAPI file for writing
k = ioT.open("~/tmp/test1.ioapi","w",ioT.iofileFlag)

## Write a list of variables to the file
k.write([co, o3, no2])

## Close the file
k.close()
```

The *write* command will produce alot of messages to stdout. Within the *write* command, the python code is calling the Fortran IOAPI library, which is actually doing the writing to the file. The stdout can be captured to a log file, see help on *ioT.open*.

Notice that the big difference between the IOAPI and the CF netCDF write is that you are doing a single write. The IOAPI file needs to be created with all of its metadata written to the file header, including the number and name of the variables in the file. If you tried to make a second write to this file, say for "so2", the *write* method would raise an error.

Note: All variables written to an IOAPI file must have the same temporal and spatial domain.

. Contents

3. Writing to an IOAPI file: multiple writes

Performing multiple writes to an IOAPI file requires an additional step:

```
## Get metadata from the variables that will
## eventually be written to the file
newM = ioT.combineMeta([o3,no2,co])

## Open IOAPI file for writing, with this metadata
j = ioT.open("~/tmp/test2.ioapi","w",ioT.iofileFlag,newM)

## Make one write to the file
j.write(no2)

## make a second write, list of variables
j.write([co,o3])

## close file
j.close()
```

Unlike the single write example, here we are explicitly extracting the metadata necessary to write all the variables, "newM". This is simply a list of an ioM object and a cdsmM object (an object which contains all the domain and axes info). In the single write example, the metadata is only written to the file during the *write* method. Here, the metadata is written to the file in the *ioT.open* function call. Because the metadata is already in the header of the file, the variables can be written at your leasure. If you try and write a variable that is not in the header (i.e. not o3, co, or no2), then the write will fail.

. . .
Contents Previous Next